universidade anhembi morumbi
Mundialmente criativa e inovadora
Laureate International Universities®

Programação para Jogos 2D

Prof. Alexandre Nascimento Tomoyose

1º semestre 2017

universidade anhembi morumbi
Mundialmente criativa e inovadora
Laureate International Universities®

Orientação a Objetos

Prof. Alexandre Nascimento Tomoyose

1º semestre 2017

Orientação a Objetos

- **Herança**
  - Conceito da Orientação a Objetos de criar uma subclasse baseada em outra classe
  - A classe base é chamada também de superclasse ou classe pai

Orientação a Objetos

- **Herança**
  - **Analogia**
    - **Felino é subclasse de Animal**
    - **Gato é subclasse de Felino**

Orientação a Objetos

- **Herança**
  - A classe filha herda atributos (variáveis) e métodos (funções) da classe pai
  - Na linguagem Java, a palavra-chave para definir uma herança é **extends**
  - No caso do C# é o "**:**" (dois-pontos)

Orientação a Objetos

- **Herança**
  - As classes filhas podem redefinir os métodos da classe pai e assim terem comportamento diferente, isso é chamado de **sobrescrita** (override)
  - Exemplo conceitual
    - A classe Felino possui um método “correr()”
    - A subclasse Gato pode ter um “correr()” próprio que sobrescreve o método da classe pai

Orientação a Objetos

- **Exercício 1**
  - **Criar uma classe Forma:**
    - **atributos: x, y, largura, altura, cor**
    - **método desenha() que escreve no console os valores de x e y**
    - **método construtor que gera valores aleatórios para x e y**
    - **método inicializa() que recebe como parâmetros os valores de: largura, altura, cor**

Orientação a Objetos

- **Exercício 1b**
  - **Criar uma classe Elipse:**
    - **subclasse de Forma**
    - **sobrescrever método desenha() para fazer uma elipse com os atributos que foram herdados de Forma**
  - **Função setup:**
    - **criar uma instância de Elipse e utilizar os métodos inicializa() e desenha()**

Orientação a Objetos

- **Exercício 1c**
  - **Criar uma classe Retangulo:**
    - **subclasse de Forma**
    - **sobrescrever método desenha() para fazer um retângulo com os atributos que foram herdados de Forma**
  - **Função setup:**
    - **criar uma instância de Retangulo e utilizar os métodos inicializa() e desenha()**

Orientação a Objetos

- **Polimorfismo**
  - Propriedade de Orientação a Objetos que decorre diretamente do conceito de herança
  - Consiste na possibilidade de usar um objeto de qualquer subclasse no lugar de uma classe base
  - Utiliza o fato das classes filhas herdarem os atributos e métodos da classe pai

Orientação a Objetos

- **Polimorfismo**
  - **Analogia**
    - **Felino é subclasse de Animal**
    - **Gato é subclasse de Felino**
  - **Portanto:**
    - **Todo Felino representa um Animal**
    - **Todo Gato representa tanto um Felino como também um Animal**

Orientação a Objetos

- **Polimorfismo**
  - Na prática, variáveis que são do Tipo de Dado de uma classe pai podem armazenar instâncias de classes filhas
  - Exemplo conceitual
    - Uma variável do tipo Felino pode armazenar um Gato
    - Uma variável do tipo Animal pode armazenar tanto um Felino quanto um Gato

Orientação a Objetos

- **Exercício 2**
  - **Classe Forma:**
    - **criar atributo pontos**
    - **modificar método construtor para atribuir o valor zero a pontos**
  - **Classe Retangulo:**
    - **criar método construtor que altera pontos para -1**
  - **Classe Elipse:**
    - **criar método construtor que altera pontos para 1**

Orientação a Objetos

- **Exercício 2b**
  - **Classe Forma:**
    - **criar método detecta() que sempre retorna falso**
  - **Classe Retangulo:**
    - **sobrescrever método detecta() para retornar verdadeiro caso o mouse esteja sobre ele**
  - **Classe Elipse:**
    - **sobrescrever método detecta() para retornar verdadeiro caso o mouse esteja sobre ela**

Orientação a Objetos

- **Exercício 2c**
  - **Criar função global geraForma():**
    - **instancia e retorna um objeto do tipo Elipse ou Retangulo com tamanho e cor aleatórias**
  - **Função draw:**
    - **gerar e desenhar uma nova instância a cada segundo (millis)**

Orientação a Objetos

- **Exercício 2d**
  - **Função draw:**
    - **criar marcador de pontuação**
  - **Criar função global somaPontos():**
    - **recebe uma Forma como parâmetro e soma os pontos dela no marcador de pontuação**

Orientação a Objetos

- **Exercício 2d**
  - **Função mousePressed:**
    - **detectar se o jogador clicou em uma Elipse ou Retangulo e chamar a função somaPontos() passando a instância clicada**
    - **limpar a tela e bloquear a detecção até que uma nova instância seja criada**

Orientação a Objetos

- **Exercício 2 - Desafios**
  - **Torne a dificuldade do jogo dinâmica, aumentando a velocidade conforme o jogador ganha pontos**